Attention is all you need

Abstract

- 좋은 성능을 보이는 모델들은 인코더와 디코더를 어텐션(Attention) 메커니즘을 통해 연결
- 어텐션 메커니즘만을 기반으로 하는 새로운 간단한 네트워크 아키텍쳐인 Transformer를 제안함
- Transformer 모델을 통해 병렬 처리에 유리하고 훈련 시간을 크게 단출할 수 있음을 확인
- 대규모 및 한정된 훈련 데이터 환경 모두에서 성공적으로 기능함

1. Introduction

- RNN, LSTM, GRU(Gate Recurrent Unit) 등은 언어 모델링 및 기계 번역과 같은 sequence 모델링 및 변환 문제에서 뛰어난 성과를 보이는 Recurrent 모델
- 순환 모델은 시간 단계(Time Step)별로 위치를 정렬하여 연산을 진행하며, 이전 은닉 상태 h_{t-1}및 현재 위치 t의 입력을 이용해 새로운 은닉 상태h_t를 생성
  -> **본질적인 순차적 처리 방식**으로 인해, 개별 학습 샘플 내에서 병렬 처리가 불가능함
  -> 시퀀스 길이가 길어질수록 메모리 제한으로 인해 여러 샘플을 동시에 배치 처리하는 것이 어려움
- 최근 연구에서 인수분해 기법 및 조건부 연산을 활용하여 연산 효율 개선, 모델 성능 향상 **but, 순차적 연산이 필요한 구조적 한계 남아있음**
- 어텐션 메커니즘은 입력 및 출력 시퀀스 내에서 거리에 관계없이 종속성을 모델링할 수 있도록 지원함
- Transformer 모델은 순환 구조를 배제하고, 오직 어텐션 메커니즘만을 활용하여 입력과 출력 간의 전역적인 종속성을 모델링, 높은 병렬 처리 기능 제공

2. Background

- 순차적 연산을 줄이려는 목표는 Extended Neural GPU, ByteNet, ConvS2S 에서도 연속적 연산을 줄이기 위한 연구가 이루어졌는데, 모두 CNN을 기본 구성 요소
  -> 입력 및 출력 위치 전체에서 병렬적으로 은닉 표현(Hidden Representations)을 계산
  -> but, 이러한 모델에서 임의의 두 입력 또는 출력 위치 간의 신호를 연결하는 데 필요한 연산량이 위치 간 거리에 따라 증가
  (ex. ConvS2S - 선형적으로 증가, ByteNet - 로그 형태로 증가)
  -> 먼 거리 종속성을 학습하는 것이 어려워지는 문제가 발생

  vs
- Transformer에서는 위의 연산량을 상수 수준으로 줄일 수 있음
  -> 어텐션 가중치가 적용된 여러 위치를 평균 내는 방식으로 인해 실제 해상도가 낮아질 위험이 존재
  -> 이 위험을 multi-head attention을 활용하여 보완

- Transformer는 시퀀스 정렬된 RNN이나 합성곱을 사용하지 않고 오직 Self-attention 만을 활용하여 입력 및 출력 표현을 계산하는 최초의 변환 모델

Self attention : 단일 시퀀스 내 서로 다른 위치 간의 관계를 학습하여 해당 시퀀스의 표현을 계산하는 어텐션 메커니즘
End-to-end memory networks : 시퀀스 정렬된 순환(RNN) 구조 대신, 순환 어텐션 메커니즘 (Recurrent Attention Mechanism)을 기반으로 동작

3. Model Architecture

3.1 Encoder and Decoder Stacks

Encoder : N개의 동일한 레이어로 구성

- 각 레이어는 **셀프 어텐션 기법**과 **피드포워드 신경망 포함
  Decoder : 인코더와 유사하지만 추가적인 Masked Self-Attention 포함
- 마스크를 적용하여 미래 토큰을 보지 않도록 제한

  3.2 Attention
  Scaled Dot-Product Attention
  : Q (Query), K (Key), V (Value) 사용하여 관계 학습
  : 스케일링($\sqrt{d_k}$)을 통해 안정적인 학습 유도
  Multi-Head Attention
  : 여러 개의 어텐션 헤드를 사용하여 다양한 관계를 학습
  : 각 헤드는 독립적인 가중치를 학습하고, 최종적으로 결합

  3.3 Position-wise Feed-Forward Networks
- 각 인코더 및 디코더 레이어에 포함된 개별 피드포워드 신경망 적용
- 두 개의 선형 변환과 ReLU 활성 함수로 구성
- 각 포지션에서 독립적으로 작용하여 비선형 변환 수행

  3.4 Embeddings and Softmax
- 입력 및 출력 토큰을 벡터로 변환하기 위해 학습된 임베딩 사용
- 디코더 출력은 선형 변환과 softmax를 거쳐 다음 토큰의 확률을 예측
- 임베딩 레이어와 소프트맥스 직전 선형 변환의 가중치 행렬을 공유

  3.5 Positional Encoding
- 모델이 순서를 인식할 수 있도록 입력 임베딩에 위치 인코딩을 추가
- 위치 인코딩은 d_{model} 차원을 가지며, 사인-코사인 함수를 사용하여 계산
- 주파수가 기하급수적으로 증가하여 상대적인 위치 정보를 효과적으로 인코딩 가능

4. Why Self-Attention

Self-Attention을 **Recurrent Layer 및 Convolution Layer**와 비교하여 세 가지 기준으로 분석

1. 연산 복잡도(Computational Complexity)
   : RNN 계열에서는 하지 못했던 병렬 처리 연산의 양을 대폭 늘려 자연스레 학습 시간 감소
2. 병렬화 가능성(Parallelization)
   : 트랜스포머 모델은 대응관계가 있는 토큰들 간의 물리적인 거리값들 중 최댓값이 다른 모델에 비해 매우 짧아 '장기간 의존성'을 잘 학습할 수 있고 시퀀스 변환 문제도 잘 해결할 수 있음
3. 장기 의존성 학습(Long-Range Dependencies)
   : 'Attention' 이라는 가중치를 시각화하여 토큰들 간의 대응관계를 눈으로 직접 확인 가능

7. Conclusion

- **Transformer 모델**을 제안하였으며, 이는 어텐션(Self-Attention) 기반으로 작동하는 최초의 시퀀스 변환(Sequence Transduction) 모델임
- 기존 인코더-디코더 구조에서 사용되던 순환 레이어(Recurrent Layers)를 multi-head self attention로 대체
- 번역(Translation) 작업에서 Recurrent Layer나 Convolution Layer 기반 아키텍처보다 훨씬 빠르게 학습 가능